

I hereby certify that this correspondence is being filed via EFS-  
Web with the United States Patent and Trademark Office  
on August 25, 2008

TOWNSEND and TOWNSEND and CREW LLP

By *Sylvia E. Arnold*  
Sylvia E. Arnold

PATENT

Attorney Docket No.: 021735-000500US

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of:

Craig Weissman et al.

Application No.: 10/817,161

Filed: April 2, 2004

For: CUSTOM ENTITIES AND FIELDS  
IN A MULTI-TENANT DATABASE  
SYSTEM

Confirmation No.

Examiner: Leon Jonathan Harper

Technology Center/Art Unit: 2166

APPELLANTS' BRIEF UNDER  
37 CFR §41.37

Mail Stop Appeal Brief  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Appellant hereby submits this Appeal Brief pursuant to 37 CFR §1.192(a). An Electronic Acknowledgement Receipt generated upon the filing of appellant's Notice of Appeal indicates that the date of receipt of appellant's notice of appeal is January 25, 2008. Thus, pursuant to 37 CFR §1.192(a), this Appeal Brief was due on March 25, 2008, extensions of time being permitted. Accordingly, Appellants request a five month extension of time to extend the due date to August 25, 2008. The Commissioner is hereby authorized to charge deposit account no. 20-1430.

## TABLE OF CONTENTS

1. REAL PARTY IN INTEREST .....	3
2. RELATED APPEALS AND INTERFERENCES.....	4
3. STATUS OF CLAIMS .....	5
4. STATUS OF AMENDMENTS .....	6
5. SUMMARY OF CLAIMED SUBJECT MATTER .....	7
6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	9
7. ARGUMENT .....	10
8. CONCLUSION.....	<b>Error! Bookmark not defined.</b>
9. CLAIMS APPENDIX.....	34
10. EVIDENCE APPENDIX.....	35
11. RELATED PROCEEDINGS APPENDIX .....	36

**1. REAL PARTY IN INTEREST**

The real party in interest of the subject patent application is Salesforce.com, Inc., the owner of the patent application.

## **2. RELATED APPEALS AND INTERFERENCES**

There are no known related appeals, interferences or judicial proceedings which may be related to, directly affect or be directly affected by or have bearing on the Board's decision in the pending appeal.

### 3. STATUS OF CLAIMS

Claims 1-26 are pending in the application.

Claims 1-4 20, 22 and 24-26 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Lin, U.S. Patent Application Publication No. US 20050071345, hereinafter referred to as "Lin".

Claims 5-19, 21 and 23 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. US 20030154197, hereinafter referred to as "Millet".

Appellants are appealing herein the rejections of claims 1-26.

**4. STATUS OF AMENDMENTS FILED SUBSEQUENT TO FINAL REJECTION**

Applicants filed an Amendment subsequent to the Final Office Action mailed July 27, 2007. No claim amendments were presented in this amendment. The amendment was considered in an Advisory Action after final rejection on December 31, 2007, but was found not to place the Application in a condition for allowance.

## **5. SUMMARY OF CLAIMED SUBJECT MATTER**

The present invention provides systems and methods for hosting variable schema data in a fixed physical database schema. (para. 0005) In one aspect, standard objects are provided in a multi-tenant database system for use by multiple organizations, and security mechanisms are provided to keep each tenant's data separate unless the data is shared (para. 0006, 0027). Each organization may add or define custom objects and custom fields, and may designate fields for indexing (para. 0006). Custom fields for multiple tenants are stored within the database structure in a single field that may contain different data types for each tenant (para. 0006). Among other advantages, embodiments enable custom entities and fields to be provided while enabling dictionary-overwhelming, maintenance burden and various problems relating to a large population of different tenants to be avoided (para. 0003).

In one aspect, a multi-tenant database system provides a standard main table 200 and an associated custom field table 210 which are shown in FIG. 3 (para 0037). Standard main table 200 includes an organization ID column 201, a table ID and any further predefined data columns 203 or standard fields provided to various organizations that might use the table. Custom field table 210 includes custom fields defined by various tenant organizations using the main table 200, which custom fields 213 are empty for an organization when it is initially associated with table 200 (para 0038, 0039). When a record or row is created in main table 200, a corresponding row is also created in custom field table 210, which allocates one of the custom field columns 213 to hold the data. As shown in Figure 4, one or more records or rows may be created for various tenant organizations 1 through N or standard or custom organization entities (given by entity divisions 460) as inter-disbursed slices of the data structure (here, blocks of rows) according to data schema that may be applicable to included tenants and/or entities. In one aspect, the system fills the lowest numbered columns first, e.g., val0 column and then val1.

Because the database is multi-tenant and a physical custom field column may contain data across multiple organizations that is not limited to specific data types, strings, numbers and dates may be found in one physical custom field column (para 0044). In one aspect, various data-types are stored as text in a canonical format that allows easy conversion back to the logical data type. In a further aspect, a custom field definition metadata table is used

to record the name, datatype and other information for each custom field column defined for each organization and table (para. 0045). Table 500, for example, includes custom\_field\_definition\_id column 510 and organization\_id column 520, as well as table name or id column 530, field name column 540, field datatype column 550, is\_indexed column 560 and column\_number column 570 (para. 0041).

While placing a functional index on each organization's slice of the data in a given custom field column is not possible from an Oracle DB perspective (e.g., because the Oracle DB does not understand a physical column containing multiple format data), indexing and other operations may nevertheless be provided (para. 0046). In one aspect, when a custom field is flagged for indexing, the data in the original column is copied to an indexed column (e.g., columns 320); in another aspect, the lowest numbered indexed columns preferably used or filled first (para. 0048). In this aspect, the original data is still maintained to display the un-modified format to the user when necessary. In a further aspect, a "case folding" algorithm converts each string custom field value to a universal case insensitive format, and processing may be conducted on the corresponding case-folded indexed column, for example, enabling the problem of searching across multiple languages to be avoided (para. 0050).

In a further aspect, a separate table is provided that contains data values for customers who require uniqueness. Ongoing changes to a custom field column are updated synchronously to the unique index table. One schema for a unique index maintenance table includes (1) organization-id, (2) custom field definition id and (3) custom field value, which allows multiple custom fields from a same organization (and entity) to be indexed (para. 0052-0055).



**6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

The issues on appeal are:

Rejection of claims 1-4, 20, 22 and 24-26 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. US 20050071345 to Lin (hereinafter "Lin").

Rejection of claims 5-19, 21 and 23 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. US 20030154197 to Millet (hereinafter "Millet").

## 7. ARGUMENT

### I. Claims 1-4 20, 22 and 24-26 are not anticipated under 35 USC §102(e) by Lin.

The initial burden of establishing a basis for denying patentability to a claimed invention rests upon the Examiner. See *In re Piasecki*, 745 F.2d 1468, 223 USPQ 785 (Fed. Cir. 1984) (emphasis added). The Examiner must show that a reference teaches “every aspect” of the subject claims “either explicitly or impliedly”, or the reference does NOT anticipate the subject claims under 102(e). MPEP 706.02 (emphasis added.) Examination is NOT, however, “open-ended” and limitations are instead imposed on the Examiner’s interpretation of the subject claims, use of a reference in establishing *prima facie* anticipation and proper review.

In interpreting claims, the claims must be “given their broadest reasonable interpretation consistent with the specification” as it would be “interpreted by one of ordinary skill in the art”, *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364, 70 USPQ2d 1827 (Fed. Cir. 2004) (emphasis added). An Applicant may, however, rebut the ordinary and customary meaning of claim terms under MPEP 2111.01. The Examiner should also construe the claim preamble “as if in the balance of the claim” if it “recites limitations” or “is ‘necessary to give life, meaning, and vitality’ to the claim”, *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165 (Fed. Cir. 1999). Emphasis is added. See also MPEP 2111.

In asserting a reference, the “identical invention must be shown in as complete detail as is contained in the... claim” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added). The Examiner may NOT, however, modify a reference to meet the claim, which “is reserved for 35 U.S.C. 103”. MPEP 706.02.

Anticipation by inference also requires more than mere Examiner assertion. In *In re Lamberti*, for example, the inference asserted by the Examiner was upheld “because the reference recognized the possibility of using temperatures greater than 750°C”, and, “700°C was much lower than had previously proved feasible”, *In re Lamberti*, 545 F.2d 747, 750, 192 USPQ 278, 280 (CCPA 1976) (emphasis added).

As with the pending application, the assertedly anticipating reference must “provide an enabling disclosure of the desired subject matter. Mere naming or description is

insufficient if it cannot be produced without undue experimentation", *Elan Pharm., Inc. v. Mayo Found. For Med. Educ. & Research*, 346 F.3d 1051, 1054, 68 USPQ2d 1373, 1376 (Fed. Cir. 2003) (emphasis added). See also MPEP 2121.01.

Additionally, while argument that the alleged anticipatory prior art teaches away from the invention "is not 'germane' to a rejection under section 102", *Twin Disc, Inc. v. United States*, 231 USPQ 417, 424 (Cl. Ct. 1986) (quoting *In re Self*, 671 F.2d 1344, 213 USPQ 1, 7 (CCPA 1982)), anticipation was found to apply IF the reference disparages the invention "after disclosing it" *Celeritas Technologies Ltd. v. Rockwell International Corp.*, 150 F.3d 1354, 1361, 47 USPQ2d 1516, 1522-23 (Fed. Cir. 1998) (emphasis added). See also MPEP 2131.05.

A degree of fairness or "duality" is also required as to Examiner review. For example, just as the Applicant must respond to all objections and rejections in order to avoid presumed admission, the Examiner must also respond to Applicant arguments, or risk allowability of the claims to which Applicant argument remains un-challenged, *In re Herrmann*, 261 F.2d 598, 120 USPQ 182 (CCPA 1958). See also *In re Soni*, 54 F.3d 746, 751, 34 USPQ2d 1684, 1688 (Fed. Cir. 1995) (Office failed to rebut applicant's argument) and MPEP 707(f). Further limitations may also apply in addition to those specifically mentioned herein.

**A. The Examiner failed to establish and Lin fails to anticipate each of the Claim 1 elements under 35 U.S.C. 102(e)**

Applicants will now consider the sole reference to Lin that was relied upon in the final rejection of the claims 1-4, 20, 22 and 24-26 under 35 U.S.C. 102(e). Applicants will also consider the sole reference to Millet, which the Examiner relied upon in a same manner in the immediately preceding non-final rejection of the same claims, and for which the Examiner again failed to elaborate, or to question or respond to similar Applicant arguments.

Claim 1 is an independent claim and claims 2-4 and 24-26 are dependent claims depending from claim 1. Claims 20 and 22 are independent claims; these claims are considered following the present consideration of independent claim 1 and dependent claims 2-4 and 24-26.

Regarding claim 1, the Examiner in the present Final Office Action first barely alleges (i.e., without elaboration) that Lin paragraph 0028 teaches

*“a data structure having a plurality of data columns and one or more index columns”.*

Applicants respectfully submit, however, that even assuming *arguendo* that the assertion may be correct (which Applicants respectfully disagree), the Examiner would nevertheless fail to establish *prima facie* anticipation under 102(c). Specifically, the Examiner’s assertion pertains to ONLY a PART of the corresponding claim 1 element, and claim 1 INSTEAD recites “defining a multi-tenant data structure having a plurality of data columns and one or more index columns” (emphasis added). Thus, applying *Richardson* and MPEP 2131, the Examiner has failed to show that Lin discloses the identical invention in as complete detail as is contained in claim 1, here ignoring clearly substantive matter. Therefore, the Examiner has also failed to establish anticipation of at least one element of claim 1, and thus failed to establish *prima facie* anticipation under 35 U.S.C. 102(c) as well.

Lin also fails to anticipate at least the element that is actually recited by claim 1: “defining a multi-tenant data structure having a plurality of data columns and one or more index columns”. As was already discussed, a multi-tenant data structure according to embodiments of the invention provides standard objects for use by multiple organizations that may share the data structure, and security to keep each tenant’s data separate unless the data is shared. Multi-tenant data structure embodiments that provide for custom entities and fields are further discussed throughout the instant title of the invention, specification, drawings and claims. Lin not only fails to anticipate the recited “defining”, but further fails entirely to disclose the recited multi-tenant data structure and fails to provide ANY basis according to which a multi-tenant database may be properly implied under MPEP 2144.01. Moreover, Lin fails to provide ANY enabling disclosure of the recited “defining” under MPEP 2121.

Lin therefore fails to anticipate claim 1 for at least these reasons.

The cited Lin paragraph 0028, for example, merely teaches “*storing... custom attributes of the same data type*” in a column of a “*custom attribute table*”, which provides NO disclosure of the recited “defining a multi-tenant data structure...” NOR any basis from which such “defining” may be reasonably implied. Moreover, the immediately following Lin

paragraph 0029 teaches that “*the number of custom-attribute tables... increases relative to the number of data types of the custom attributes*”. Therefore, applying the recited “defining a multi-tenant data structure...” in Lin would require a huge number of custom attribute tables to accommodate the multiplicity of applications of multiple organizations, and groups and users within the different tenant organizations, thereby rendering Lin NOT operable for practicable use and therefore NOT patentable under 35 U.S.C. 101. Applicants respectfully submit that one of ordinary skill in the art at the time of the invention clearly would NOT infer that Lin intended a NOT operable and NOT patentable result. Lin also fails to provide an enabling disclosure under MPEP 2121.01, since one of ordinary skill in the art would NOT be alerted to, let alone provided with any basis whatsoever for recognizing or resolving the resulting in-operability of Lin, let alone further modification necessitated by the “recited defining”, -at least without extensive and thus clearly undue experimentation. The remainder of Lin similarly fails to anticipate the recited “defining a multi-tenant data structure...” for at least the same reasons.

Therefore, *prima facie* anticipation of claim 1 over Lin simply does not exist for these reasons as well.

The Examiner next barely alleges, respecting claim 1, that Lin paragraph 0028 ALSO discloses BOTH of:

*“defining a first data field for a first tenant, said first field having a first data type”,*

and

*“defining a second data field for a second tenant, said second field having a second data type, wherein the second data type may be different than said first data type”.*

Applicants respectfully submit, however, that as discussed respecting the above “defining” element, Lin fails entirely to disclose a multi-tenant data structure. Lin further fails to not only expressly disclose, but also to even consider any aspect of tenancy, let alone a “first tenant”, a “second tenant”, or further, either of the recited “*defining a first data field for a first tenant...*”

and “*defining a second data field for a second tenant...*”. There is also no proper basis in Lin from which to draw an inference as to these elements, as required under MPEP 2144.01, either from Lin paragraph 0028 (“storing... custom attributes of the same data type” in a column of a “custom attribute table”) or elsewhere in Lin.

For example, Lin discloses NO mechanism whatsoever that may provide for identifying users or user groups as corresponding to different tenant organizations, NO mechanism for processing particular tenant data and NO mechanism such that “one tenant does not have access to another’s data, unless such data is expressly shared” (instant specification at paragraph 0023). Moreover, Lin clearly neither discloses nor provides ANY reasonable basis for inferring the recited “defining a first data field for a first tenant, said first field having a first data type” or the recited “defining a second data field for a second tenant, said second field having a second data type, wherein the second data type may be different than said first data type”. Lin also fails to provide any disclosure that may be reasonably construed as an enabling disclosure such that these elements may be implemented by one of ordinary skill in the art without undue experimentation (MPEP 2121).

Therefore, not only has the Examiner failed to establish *prima facie* anticipation under 102(e), but Lin further fails to anticipate claim 1 for at least these reasons as well.

The Examiner also barely alleges, respecting claim 1, that Lin paragraph 0031 discloses the claim 1 element:

*“when records having data values in the first and second fields are created by the first and second tenants, storing the data values of first and second fields to a single column in the data structure, wherein the single column includes data values that may include different data types for different tenants”.*

Applicants respectfully submit, however, that Lin not only fails to anticipate the recited element, but further repeatedly contradicts the Examiner’s assertion. Lin paragraph 0031, for example, discloses that a custom-attribute table includes one or more instance-identifying columns,

attribute-identifying columns and value columns, each row holding data that is associated with a particular object instance of an object type. The immediately following Lin paragraphs 0032 - 0034, however, disclose that “*the value column[s]... provide storage for values that are of the type... associated with the custom-attribute table”.* Lin further discloses that: “*a single column is provided for storing data for all custom attributes of the same data type”*, “*as long as the custom attributes are of the same data type...*” and “*where the first and second custom attributes have the same data type”* (previously cited paragraph 0028). See also Lin paragraphs 0032 (“... storage for values that are of the data type”), 0044 (“... stores values for all custom attributes of that data type”), as well as at least Lin figures 1 and 2. Moreover, Lin barely contradicts the Examiner, without ever considering a single column including data values that may include “different data types”, or further, “different types for different tenants”, and substantial undue experimentation would clearly be required to do so. Therefore, not only has the Examiner failed to establish *prima facie* anticipation under 102(e), but Lin further fails to anticipate claim 1 for at least these reasons as well.

**B. The Examiner failed to establish and Lin fails to anticipate  
each of the Claim 1 elements under MPEP 2144.01**

While the Examiner failed to elaborate or to question Applicants’ Arguments during normal responsive prosecution, the Examiner finally responded in some manner in an Advisory Action following Applicants’ Response to the instant Final Office Action. Applicants first respectfully submit, however, that the included Examiner assertions address only an Examiner-selected subset of Applicants’ arguments further respecting only Claim 1, and NOT the remaining claim 1 or other of Applicants’ Arguments. (See item III below.) Moreover, the Examiner’s assertions not only fail to establish *prima facie* anticipation, but even assuming arguendo that the Examiner may be correct (with which Applicants respectfully disagree), the Examiner nevertheless failed to establish *prima facie* anticipation under 102(e).

Specifically, the Examiner ADMITS that Lin fails to expressly disclose a multi-tenant database (“may not explicitly use the term... multi-tenant database”). Applicants presume that the Examiner refers to the “multi-tenant data structure” as “defined” in to claim 1. The

Examiner must therefore establish that Lin “impliedly” teaches the respective claim 1 elements or Lin does NOT anticipate the subject claims under 102(e). MPEP 706.02 (emphasis added.) The remaining limitations for claim interpretation and anticipation, including but not limited to those given above, must also be met in order for *prima facie* anticipation to apply.

The Examiner asserts that Lin implicitly discloses the “defining...” elements of claim 1 MPEP 2144.01 responsive to the following particularly reiterated subset of Applicants’ Arguments:

*(that) the Examiner failed to consider the recited “defining a multi-tenant data structure...”, whether Lin anticipates a multi-tenant data structure, that Lin fails to do so, that such failure caused the Examiner ALSO to fail to consider “defining a first tenant...”, “defining a second tenant...” or to incorrectly associate “tenant” organization with user.*

In applying MPEP 2144, however, the Examiner characterizes the “essence” of the “functionality” of a multi-tenant database as “allowing multiple customer organizations to share database resources”. The Examiner then asserts the “custom attributes” in Lin “allow” such sharing and one skilled in the art would therefore infer such from reading Lin. Applicants respectfully disagree.

Contrary to the Examiner’s unsupported “essence” assertion, claims must be “given their broadest reasonable interpretation consistent with the specification” as it would be “interpreted by one of ordinary skill in the art”, *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364, 70 USPQ2d 1827 (Fed. Cir. 2004) (emphasis added), and an Applicant may further be his own lexicographer under MPEP 2111.01(iv). Under either standard, the instant specification clearly discloses that a multi-tenant data structure provides for keeping each tenant’s data separate unless the data is shared. Moreover, claim 1 recites processing that may be conducted differently for different tenants, e.g., including “*defining a first data field for a first tenant...*” and “*defining a second data field for a second tenant*”).

Contrastingly the cited Lin paragraph 0026 merely summarily discloses that “techniques are provided to allow users to store data for an unlimited number of custom



attributes... without adding any new columns to the existing tables...". (Lin also discloses allowing performing upgrades to an application, allowing custom object types and attributes data retrieval, and storing custom attribute identifiers in memory.) Such disclosure, as with the remainder of Lin, therefore fails to provide ANY "recognition" or ANY other indicia of the recited "multi-tenant data structure", let alone the level of indicia required for establishing implied disclosure under *In re Lamberti*, 545 F.2d 747, 750, 192 USPQ 278, 280 (CCPA 1976) (emphasis added).

Further, Lin is directed at "techniques... to allow users to store data for a number of custom attributes of application object types" (Lin Abstract). Thus, contrary to the Examiner's assertion, one of ordinary skill would NOT reasonably infer, from indicia in Lin, that: Lin's application object type attributes INSTEAD identify user, group and entity tenancy, the data structure is also MODIFIED prior to deployment to account for all potential tenants, Lin is MODIFIED to operate according to such tenancy, Lin is MODIFIED to identify shared and not shared data, Lin is MODIFIED to prevent a user from entering/changing custom user data as taught by Lin, such NOT shared data may not be accessed or modified, and so on – which is impermissible under MPEP 706.02. Moreover, the cited paragraph and the remainder of Lin are entirely devoid of ANY recognition or further basis for inferring the recited "multi-tenant data structure" under *In re Lamberti*, 545 F.2d 747, 750, 192 USPQ 278, 280 (CCPA 1976), and Lin fails to provide an enabling disclosure under MPEP 2121.01. Additionally, the present invention provides advantages such as enabling dictionary-overwhelming, maintenance burden and other problems relating to a large population of different tenants to be avoided. Such advantages are NOT present in Lin and asserting a multi-tenant data base would instead cause Lin to become NOT operable, as was already discussed.

Even assuming arguendo that the Examiner's assertions in the Advisory Action may be correct (with which Applicants respectfully disagree), the Examiner would nevertheless have failed to establish prima facie anticipation and Lin fails to anticipate claim at least because Lin fails to anticipate the recited "*when records having data values in the first and second fields are created by the first and second tenants, storing the data values of first and second fields to a single column in the data structure, wherein the single column includes data values that may*

include different data types for different tenants". This argument further remains un-questioned by the Examiner.

Therefore, for at least the foregoing reasons, Applicants respectfully submit that the Examiner has failed to establish anticipation by inference and Lin clearly does NOT anticipate Claim 1. Applicants further submit that the Examiner has failed to establish and Lin fails to impliedly or otherwise anticipate claim 1 for the reasons already given respecting individual claim 1 elements as well.

**C. The Examiner failed to establish and Lin fails to anticipate  
Claims 2-4 and 24-26 under 35 U.S.C. 102(e)**

Claims 2 - 4 and 24 - 26 are dependent claims depending from claim 1. Therefore, the Examiner has also failed to establish *prima facie* anticipation and Lin fails to anticipate claims 2-4 and 24-26 for at least the same reasons as set forth for claim 1. Applicants further respectfully submit the following respecting claims 2-4 and 24-26.

The Examiner barely alleges, respecting claim 2, that Lin paragraph 0037 discloses:

*"defining a separate data structure having one or more columns; and in response to an indication from one of the first tenant and the second tenant that data in the first data field or second data field, respectively, be unique, copying the data values stored in the single data column corresponding to the first data field or second data field, respectively, to a column in the separate data structure" (emphasis added).*

Applicants respectfully submit, however, that Lin paragraph 0037 fails entirely to expressly or impliedly even relate to "uniqueness", let alone the recited responding to a uniqueness "indicator" by "copying data values... to a column in" a "separate data structure". Lin paragraph 0037 also fails entirely to anticipate the recited copying of "data values stored in" a "single data column" to a "column in the separate data structure".

The cited Lin paragraph, for example, INSTEAD relates to conducting an “upgrade of” an “application”. Moreover, Lin’s application upgrade INSTEAD provides for: “copying... data... stored in [a] first table” to a “first replacement table”, “deleting the first table”, “copying... data... stored in [a] second table” to a “second replacement table” and “deleting the second table”, both of which are entirely un-related to the embodiment of claim 2. Moreover, the cited Lin paragraph and the remainder of Lin ALSO fail entirely to anticipate the tenancy, let alone the recited responding “to an indication from one of a first tenant and a second tenant”, for at least the same reasons as argued respecting claim 1. Applicants therefore respectfully submit that Lin fails to anticipate claim 2 and the Examiner failed to establish *prima facie* anticipation under 35 U.S.C. 102(e) for at least the foregoing reasons.

The Examiner also barely alleges, respecting claim 3 and dependent claims 4, 25 and 26 that depend from claim 3, that Lin paragraph 0040 discloses:

“copying to a first one of the index columns the data values stored in the single data column for the first field in response to a request from the first tenant to index data in the first data field” (emphasis added).

Applicants respectfully submit, however, that Lin paragraph 0040 fails entirely to expressly or impliedly even relate to “indexing” let alone “copying data... in response to a request to index data”, “copying... data values stored in the single data column” or responding “to a request from the first tenant”.

The cited Lin paragraph, for example, INSTEAD relates to Lin’s continuing discussion of the application upgrading submitted in conjunction with claim 2. Specifically, Lin provides, during application upgrading in which “the structures of second and third tables are not modified”, for retaining “the data values for the first custom attribute in the second table” and “the data values for the second custom attribute... in the third table”, which is entirely unrelated to indexing, let alone indexing in conjunction with a multi-tenant data structure. Moreover, the cited Lin paragraph and the remainder of Lin ALSO fail entirely to anticipate

tenancy, let alone the recited responding “to a request from the first tenant”, for at least the same reasons as argued respecting claim 1.

Regarding claim 4, the Examiner further barely asserts that Lin paragraph 0071 discloses the recited “converting the copied data values” (that are “stored in the single data column” and copied “in response to a request from the first tenant to index data” in claim 3) “to a modified format”. Applicants respectfully submit, however, that the further cited Lin paragraph, which is completely unrelated to Lin paragraph 0040 (above), again fails entirely to expressly or impliedly even relate to “indexing”. Lin paragraph 0071 INSTEAD continues a discussion of a “computer system... for implementing the techniques discussed herein” that begins at Lin paragraph 0068. Lin paragraph 0071 more specifically discusses “various forms of computer readable media” that “may be involved in carrying one or more sequences of... instructions” to a “processor 404 for execution”, including a discussion of a modem, an infra red detector, main memory and a “storage device 410”, which is entirely unrelated to “indexing”, let alone indexing in conjunction with a multi-tenant data structure. Moreover, the cited Lin paragraph and the remainder of Lin ALSO fail entirely to anticipate tenancy, let alone the recited responding “to a request from the first tenant”, for at least the same reasons as argued respecting claims 1 and 3.

Regarding claim 25, the Examiner barely asserts that Lin paragraph 0071, which was cited in conjunction with instant claim 4, ALSO discloses the recited “wherein the converting” (to a “modified format”, “the copied data values” that are “stored in the single data column” and copied “in response to a request from the first tenant to index data” of claims 3 and 4) “includes applying a case folding algorithm to the data values”. Applicants respectfully submit, however, that the already discussed Lin paragraph 0071 fails entirely to anticipate “applying a case folding algorithm”, let alone the recited “converting” that “includes applying a case folding algorithm to the data values”. Moreover, the cited Lin paragraph and the remainder of Lin ALSO fail entirely to anticipate tenancy, let alone the further limitations of claims 1, 3 and 4, for at least the same reasons as argued respecting claims 1, 3 and 4.

Regarding claim 26, the Examiner barely asserts that Lin paragraph 0071, which was cited in conjunction with instant claims 4 and 25, ALSO discloses the further recited limitation of “*wherein the modified format*” of claim 4, “*comprises a common data type corresponding to the index column*”. Applicants respectfully submit, however, that Lin’s discussion of “computer readable media” in “computer system... for implementing the techniques discussed herein” simply does NOT anticipate the recited embodiment of claim 26. Moreover, the cited Lin paragraph and the remainder of Lin ALSO fail entirely to anticipate tenancy, let alone the further limitations of claims 1, 3 and 4, for at least the same reasons as argued respecting claims 1, 3 and 4.

Applicants therefore respectfully submit that Lin fails to anticipate claims 2-4 and 24-26, and the Examiner failed to establish *prima facie* anticipation under 35 U.S.C. 102(e) for at least the foregoing reasons.

**D. The Examiner failed to establish and Lin fails to anticipate  
Claims 20 and 22 under 35 U.S.C. 102(e)**

Claims 20 and 22, while independently patentable from claim 1, were rejected as “comprising essentially the same limitations as claim 1 and “for the same reasons as claim 1. Applicants therefore submit that the Examiner failed to establish *prima facie* anticipation and Lin fails to anticipate claims 20 and 22 for at least the same reasons as set forth respecting claim 1.

It is therefore respectfully submitted that the Examiner has failed to establish *prima facie* anticipation of claims 1-4 20, 22 and 24-26 and that claims 1-4 20, 22 and 24-26 are patentable over Lin for at least the foregoing reasons. Withdrawal of the rejections and early allowance of claims 1-4 20, 22 and 24-26 is respectfully requested.

**II. Rejection of claims 5-19, 21 and 23 under 35 USC §103(a) over Millet.**

Claim 5 is an independent claim and claims 6-8 are dependent claims depending from claim 1. Claim 9 is an independent claim and claims 10-19 are dependent claims depending from claim 1. Each of claims 5-8, 6-19, 21 and 23 was barely rejected under 35 USC 103(a) over Millet without further elaboration or response to Applicants' Arguments, and none was the subject of the aforementioned Advisory Action. .

Applicants respectfully submit that the Examiner failed to establish *prima facie* anticipation of claims 5-8, 6-19, 21 and 23 by Millet because the Examiner failed to show at least that Millet teaches every aspect of the claims, either explicitly or impliedly, in as complete detail as the claims under MPEP 706.02 or that Millet provides an enabling disclosure under MPEP 2121.01. Millet further fails to anticipate the claims at least because Millet does not consider, let alone teach every aspect of the claims as required under MPEP 706.02 and MPEP 2121.01.

**A. The Examiner failed to establish and Millet fails to anticipate  
Claims 5, 9, 21 and 23 under 35 U.S.C. 102(e)**

Regarding claim 5, the Examiner first barely alleges that Millet paragraph 0042 teaches

*“defining a multi-tenant data structure having a primary key column”*

Applicants respectfully submit, however, that Millet paragraph 0042 and accompanying FIGS 1 and 2 merely teach that data records directed to a single organization may be organized according to “a unique identifier, such as an employee ID” and that data corresponding to the employee-IDs may be stored in successive columns “*such as the Employee ID 1... first name 2, last name 3, job title 4...*” and in FIG. 2, ‘title’ 10, ‘WorkPhone’ 11, and ‘extension’ 12”. The remainder of Millet is further in accordance with paragraph 0042.

While Millet teaches, in paragraph 0044, that the “employee ID” may serve as a primary key, as with Lin, neither the cited paragraph nor the remainder of Millet considers let alone expressly discloses the recited “*defining a multi-tenant data structure*”. Moreover, the mere employee ID in Millet, would be implied in its conventional sense, as it is commonly used

in conventional single-organization databases, and would not provide for or imply multi-tenant database operation.

Millet fails, for example, in the cited paragraph or elsewhere, to expressly or impliedly disclose any mechanism whatsoever to determine or operate in accordance with the tenancy of a user or to keep each tenant's data separate unless the data is shared, among other deficiencies. Moreover, one skilled in the art would NOT reasonably imply, or be able to produce the recited elements without undue experimentation, as required respectively by MPEP 706.02 and *in re Lamberti*, 545 F.2d 747, 750, 192 USPQ 278, 280 (CCPA 1976), and further, MPEP 2111.01.

The Examiner also barely alleges, respecting claim 5, that Millet paragraph 0054 teaches the "multi-tenant data structure", so defined, as (also) having

*"an organization id column and a plurality of data column;" as well as "defining a first table for a first tenant, said first table having a first data field, and said first tenant having a first tenant id"; "assigning a first table id to the first table" and "defining a second table for a second tenant, said second table having a second data field, and further, said second tenant having a second tenant id; assigning a second table id to the second table; wherein when records are created for the first table by the first tenant, for each created record" (emphasis added).*

Applicants respectfully submit, however, that Millet paragraph 0054 and the corresponding Millet FIG. 5 merely disclose how the above discussed employee IDs or other "primary or foreign keys of [a] data compilation are separated into a separate data table" that only stores such keys "for a particular data compilation, such as... 'Employees', and how "the primary key cannot be used for the "Row ID...". Not only does paragraph 0054 and the remainder of Millet fail to consider, let alone suggest the recited "*defining a multi-tenant data structure*", "*organization id column*", a "first tenant", "first tenant id", "second tenant", "second tenant id" or the more particular respective recitations. Additionally, Millet paragraph 0054 disagrees with

the Examiner's assertion by teaching that the Employee ID cannot be used to identify rows belonging to a particular employee, let alone to identify or conduct operation in accordance with a particular organization tenant (as the Examiner appears to assert).

Moreover, applying a multi-tenant data structure to Millet's use of multiple tables for even multiple compilations of a single organization using a single application may, as with Lin, well require a huge number of custom attribute tables to accommodate the multiplicity of applications of multiple organizations, and groups and users within the different tenant organizations, thereby rendering Millet NOT operable for practicable use and therefore NOT patentable under 35 U.S.C. 101. Applicants respectfully submit that one of ordinary skill in the art at the time of the invention clearly would NOT infer that Lin intended a NOT operable and NOT patentable result. Lin also fails to provide an enabling disclosure under MPEP 2121.01, since one of ordinary skill in the art would NOT be alerted to, let alone provided with any basis whatsoever for recognizing or resolving the resulting in-operability of Millet, let alone further modification necessitated by the "recited defining", -at least without extensive and thus clearly undue experimentation. The remainder of Millet similarly fails to anticipate the recited "defining a multi-tenant data structure..." and defining a multi-tenant data structure "having... an organization column" for at least the same reasons.

The Examiner further barely alleges, respecting claim 5, that Millet paragraph 0055 and the corresponding FIGS. 7-9 teach

*"a) storing the value of the first data field to a single data column in the data structure; b) storing the first tenant id in the organization id column; and c) storing the first table id to the primary key column; and wherein when records are created for the second table by the second tenant, for each created record: a) storing the value of the second data field to said single data column in the data structure; b) storing the second tenant id in the organization id column; and c) storing the second table id to the primary key column; and wherein the first and second tables of the first and second tenants are stored in the data structure"* (emphasis added).



Applicants respectfully submit, however, that Millet paragraph 0055 and the corresponding Millet FIGS. 7-9 merely disclose that, “where multiple compilations are stored in the same RDBMS” (relational database management system) Millet includes “a Custom Fields Tables” data table... which correlates a... Table ID... with each particular data compilation that has a name”. Millet paragraph 0055 does NOT, as the Examiner asserts, either expressly or impliedly consider, let alone disclose at least the recited “first tenant” or “second tenant” storing a “first tenant id” or “second tenant id” or an “organization id column” either alone or as recited.

Moreover, Millet paragraph 0055 and the remainder of Millet also fail to teach “storing the value of the first data field to a single data column” or “storing the value of the second data field to said single data column”. As was discussed with reference to Lin, in accordance with recited “defining a multi-tenant data structure...”, data values from different tenants, which tenants may utilize different applications, data types and schema, may be stored in a single data column. Contrastingly, Millet, as with Lin, teaches defining columns for a single organization that utilizes different columns for storing different data and different data types; Millet further stores data from different compilations in different tables, as was already discussed. Therefore, as discussed respecting Lin and as shown in Millet FIG. 6, both define each data column, in a conventional manner, as corresponding to a singular data type and with other whole column constraints corresponding to the single organization. Therefore, neither discloses, nor would one of ordinary skill in the art imply from either reference, storing different data (here of different tenants) in a single column. Not surprisingly then, neither may be reasonably construed to provide an enabling disclosure such that one of ordinary skill to produce the recited embodiment without extensive undue experimentation.

Claims 9, 21 and 23, while independently patentable from claim 5, were rejected as “comprising essentially the same limitations as claim 1 and “for the same reasons as claim 5. Applicants therefore submit that the Examiner failed to establish prima facie anticipation over Millet and Millet fails to anticipate claims 9, 21 and 23 for at least the same reasons as set forth respecting claim 5.

Therefore, Applicants respectfully submit that the Examiner failed to establish prima facie anticipation of claims 5, 9, 21 and 23, and Millet fails to anticipate claims 5, 21 and 23 for at least the foregoing reasons.

**B. The Examiner failed to establish and Millet fails to anticipate**

**Claims 6-8 under 35 U.S.C. 102(e)**

Claims 6-8 are dependent claims depending from claim 5. Therefore, the Examiner has also failed to establish *prima facie* anticipation and Millet fails to anticipate claims 6-8 for at least the same reasons as set forth for claim 5. Applicants further respectfully submit the following respecting claims 6-8.

The Examiner barely alleges, respecting claim 6, that Millet paragraph 0055 further discloses:

*“wherein the data structure includes one or more index columns, the method further comprising: copying to a first one of the index columns the data values stored in the single data column for the first table in response to a request from the first tenant to index data in the first data field”*  
(emphasis added).

Applicants respectfully submit, however, that Millet paragraph 0055 fails entirely to anticipate at least the recited “single data column” and “the first tenant” as was already discussed respecting claim 5. Millet paragraph 0055 and the remainder of Millet further simply do NOT consider, let alone expressly or impliedly disclose at least “*copying to a first one of the index columns*”, such copying of “*the data values stored in the single data column*” or such copying “*in response to a request from the first tenant*”.

The cited Millet paragraph, for example, discloses using the aforementioned Custom Field Tables data table “*when serving a query of a particular user input screen to identify the data tables associated with that screen*” (emphasis added). Millet paragraphs 0051,

0054 and 0071 also disclose the term “indexing”. However, mere naming or description is insufficient if it cannot be produced without undue experimentation”, *Elan Pharm., Inc. v. Mayo Found. For Med. Educ. & Research*, 346 F.3d 1051, 1054, 68 USPQ2d 1373, 1376 (Fed. Cir. 2003) (emphasis added). See also MPEP 2121.01. Applicants submit that it cannot. Millet merely discusses disaggregation of data to allow a user to “create custom fields in the database without modifying the database structure” and use of primary and foreign keys as indices to identify records. Not only does Millet fail to anticipate copying data values to an index column, but Millet also fails to anticipate, expressly or impliedly, conducting copying of data within a multi-tenant or any other data structure in response to a request from a user, let alone in response to a request to index or such a request from a tenant. Therefore, Applicants respectfully submit that the Examiner failed to establish prima facie anticipation of claim 6 and Millet fails to anticipate claim 6 for at least the foregoing reasons.

Claim 7 is a dependent claim that depends from claim 6 as well as independent claim 5. Therefore, Applicants respectfully submit that the Examiner has also failed to establish *prima facie* anticipation and Millet fails to anticipate claims 7 for at least the same reasons as set forth for claim 6.

The Examiner barely alleges, respecting claim 8, that Millet paragraph 0071 further discloses:

*“The method of claim 5, wherein said first data field has a first data type, and wherein said second data field has a second data type different from the first data type, such that said single data column includes data values having said first and second data types”* (emphasis added).

The cited Millet paragraph, however, discloses a flow of operation conducted by Millet in conjunction with user querying, and NOT the storing of first and second data types that are different in a single data column. In summary, Millet paragraph 71 discloses that “the user selects [from a received HTML document] a querying option” and “a particular record in the

HTML document"... "the web server receives the primary key and determines the 'Row ID' information". "Then, in a loop... for each 'Field ID' in the 'Custom Fields Table' data table associated with the" Table ID... the web server retrieves from RDBMS "the information from the Custom Fields Table"... obtains the corresponding data field record from the 'Custom Fields Values' data table associated with each 'Field ID' and 'Row ID'... and packages the field value information for transmission..."

Applicants respectfully submit that Millet paragraph 0071 and the remainder of Millet simply fails to anticipate at least the recited "said single data column includes data values having said first and second [different] data types." Therefore, the Examiner has failed also failed to establish *prima facie* anticipation and Millet fails to anticipate claims 8 for this reason as well.

**C. The Examiner failed to establish and Millet fails to anticipate  
Claims 10-19 under 35 U.S.C. 102(e)**

Claims 10-19 are dependent claims depending from claim 9. Therefore, the Examiner has also failed to establish *prima facie* anticipation and Millet fails to anticipate claims 10-19 for at least the same reasons as set forth for claim 9. Applicants further respectfully submit the following respecting claims 10-19.

The Examiner barely alleges, respecting claim 13, that Millet paragraph 0071 further discloses:

*"The method of claim 9, wherein the data structure includes one or more index columns, the method further comprising: copying to a first one of the index columns the data values stored in the single data column for the first table in response to a request from the first tenant to index data in the first data field"* (emphasis added).

Applicants respectfully submit, however, that Millet paragraph 0071 fails entirely to anticipate at least the recited "single data column" and "the first tenant" as was already discussed respecting

claim 9. Millet paragraph 0071 and the remainder of Millet further simply do NOT consider, let alone expressly or impliedly disclose at least *“copying to a first one of the index columns”*, such copying of *“the data values stored in the single data column”* or such copying *“in response to a request from the first tenant...”*.

As discussed with reference to claim 7, the cited Millet paragraph discloses a flow of operation conducted by Millet in conjunction with user querying, and NOT the recited responding to a request, or further, a *“request from the first tenant...”*. Also discussed is that while Millet paragraphs 0051, 0054 and 0071 also disclose the term “indexing”, the similarity is in name only and insufficient under *Elan Pharm., Inc. v. Mayo Found. For Med. Educ. & Research*, 346 F.3d 1051, 1054, 68 USPQ2d 1373, 1376 (Fed. Cir. 2003) and MPEP 2121.01. Applicants submit that it cannot. Millet merely discusses disaggregation of data to allow a user to “create custom fields in the database without modifying the database structure” and use of primary and foreign keys as indices to identify records. Not only does Millet fail to anticipate copying data values to an index column, but Millet also fails to anticipate, expressly or impliedly, conducting copying of data within a multi-tenant or any other data structure in response to a request from a user, let alone in response to a request to index or such a request from a tenant. Therefore, Applicants respectfully submit that the Examiner failed to establish prima facie anticipation of claim 13 and Millet fails to anticipate claim 13 for at least the foregoing reasons.

Regarding claim 14, the Examiner further barely asserts that Millet paragraph 0070 discloses the recited *“converting the copied data values”* (that are *“stored in the single data column”* and copied *“in response to a request from the first tenant to index data”* in claim 13) *“to a modified format”*. Applicants respectfully submit, however, that the cited Millet paragraph, fails entirely to expressly or impliedly even relate to “indexing”. Millet paragraph 0070 INSTEAD relates to a “dynamic column” result purportedly achieved by Millet whereby *“to a user, a database... looks like a traditional data table except that the user appears to have control over the creation and deletion of columns in the database”*. Nowhere in paragraph 0070 does Millet consider converting copied data values in conjunction with indexing, let alone the

embodiment recited by claim 14. Applicants therefore respectfully submit that, in addition to the reasons set forth in claim 13, the Examiner failed to establish *prima facie* anticipation of claim 14 and Millet fails to anticipate claim 14 for at least the foregoing reasons.

Regarding claim 15, the Examiner further barely asserts that Millet paragraph 0071 further discloses the recited “*method of claim 14, wherein converting includes applying a case folding algorithm to the data values*”. Applicants respectfully submit, however, that Millet paragraph 0071 fails entirely to anticipate at least the recited “case folding algorithm”, let alone in accordance with the discussed claim 15 dependencies. As discussed with reference to claims 7 and 14, the cited Millet paragraph discloses a flow of operation conducted by Millet in conjunction with user querying, which is NOT at all related to the recited “*wherein the converting includes applying a case folding algorithm*”. Therefore, Applicants respectfully submit that the Examiner failed to establish *prima facie* anticipation of claim 15 and Millet fails to anticipate claim 15 for at least the foregoing reasons.

It is therefore respectfully submitted that the Examiner has failed to establish *prima facie* anticipation of claims 5-19 21 and 23 and claims 5-19 21 and 23 are patentable over Millet for at least the foregoing reasons.

### III. Conclusion

Applicants respectfully submit that the Examiner has failed to establish *prima facie* anticipation of claims 1-26 over Lin and Millet and claims 1-26 are not anticipated by Lin and Millet for at least the foregoing reasons. Therefore, withdrawal of the rejections is respectfully requested.

Please deduct the requisite fee, pursuant to 37 CFR § 1.117©, of \$340.00 from deposit account 20-1430 and any additional fees associated with this Brief.

Appl. No. 10/817,161

Further to Notice of Appeal mailed October 26, 2007

PATENT

Respectfully submitted,

A handwritten signature in cursive script, reading "Daryl C. Josephson", with a long horizontal stroke extending to the right.

Daryl C. Josephson

Reg. No. 37,365

Daryl C. Josephson, Esq.

c/o Jerald T. Gray

TOWNSEND and TOWNSEND and CREW LLP

Two Embarcadero Center, Eighth Floor

San Francisco, California 94111-3834

Tel: 650-326-2400

Fax: 650-326-2422

61338060 v1

## CLAIMS APPENDIX

1. (Previously Presented) A computer-implemented method of storing multiple fields for multiple tenants in a single multi-tenant data structure, comprising:

defining a multi-tenant data structure having a plurality of data columns and one or more index columns;

defining a first data field for a first tenant, said first field having a first data type;

defining a second data field for a second tenant, said second field having a second data type, wherein the second data type may be different than said first data type; and

when records having data values in the first and second fields are created by the first and second tenants, storing the data values of first and second fields to a single column in the data structure, wherein the single column includes data values that may include different data types for different tenants.

2. (Original) The method of claim 1, further comprising:

defining a separate data structure having one or more columns; and

in response to an indication from one of the first tenant and the second tenant that data in the first data field or second data field, respectively, be unique, copying the data values stored in the single data column corresponding to the first data field or second data field, respectively, to a column in the separate data structure.

3. (Previously Presented) The method of claim 1, further comprising copying to a first one of the index columns the data values stored in the single data column for the first field in response to a request from the first tenant to index data in the first data field.

4. (Previously Presented) The method of claim 3, wherein the copying includes converting the copied data values to a modified format.

5. (Original) A computer-implemented method of hosting multiple tables for one or more organizations in a single multi-tenant data structure, comprising:

defining a multi-tenant data structure having a primary key column, an organization id column and a plurality of data columns;



defining a first table for a first tenant, said first table having a first data field, and said first tenant having a first tenant id;

assigning a first table id to the first table;

defining a second table for a second tenant, said second table having a second data field, and said second tenant having a second tenant id;

assigning a second table id to the second table;

wherein when records are created for the first table by the first tenant, for each created record:

a) storing the value of the first data field to a single data column in the data structure;

b) storing the first tenant id in the organization id column; and

c) storing the first table id to the primary key column; and

wherein when records are created for the second table by the second tenant, for each created record:

a) storing the value of the second data field to said single data column in the data structure;

b) storing the second tenant id in the organization id column; and

c) storing the second table id to the primary key column; and

wherein the first and second tables of the first and second tenants are stored in the data structure.

6. (Original) The method of claim 5, wherein the data structure includes one or more index columns, the method further comprising:

copying to a first one of the index columns the data values stored in the single data column for the first table in response to a request from the first tenant to index data in the first data field.

7. (Original) The method of claim 6, wherein copying includes identifying the data values to be copied based on the first tenant id, the first table id and the first data field.

8. (Original) The method of claim 5, wherein said first data field has a first data type, and wherein said second data field has a second data type different from the first data type, such that said single data column includes data values having said first and second data types.

9. (Original) A computer-implemented method of storing multiple tables for one or more tenants in a single data structure, comprising:

defining a data structure having a primary key column, an organization id column and a plurality of data columns;

defining a first table for a first tenant, said first table having a first data field, said first data field having a first data type, and said first tenant having a first tenant id;

assigning a first table id to the first table;

defining a second table for the first tenant, said second table having a second data field, said second data field having a second data type different from the first data type;

assigning a second table id to the second table;

wherein when records are created for the first table, for each created record:

a) storing the value of the first data field to a single data column in the data structure;

b) storing the first tenant id in the organization id column; and

c) storing the first table id to the primary key column; and

wherein when records are created for the second table, for each created record:

a) storing the value of the second data field to said single data column;

b) storing the first tenant id in the organization id column; and

c) storing the second table id to the primary key column;

wherein the first and second tables of the first tenant are stored in the data structure, and wherein said single data column includes data values having said first and second data types.

10. (Original) The method of claim 9, further comprising:

defining a third table for a second tenant, said third table having a third data field, said third data field having a third data type, and said second tenant having a second tenant id; and

assigning a third table id to the third table;

wherein when records are created for the third table, for each created record:

storing the value of the third data field to said single data column in the data structure;

storing the second tenant id in the organization id column; and

storing the third table id to the primary key column;

wherein the first, second and third tables are stored in the data structure, and wherein said single data column includes data values having said first and second data types and said third data type.

11. (Original) The method of claim 9, wherein the first and second table ids are different.

12. (Original) The method of claim 10, wherein the first and second table ids are different, and wherein the third table id is the same as one of the first and second table ids.

13. (Original) The method of claim 9, wherein the data structure includes one or more index columns, the method further comprising:

copying to a first one of the index columns the data values stored in the single data column for the first table in response to a request from the first tenant to index data in the first data field.

14. (Original) The method of claim 13, wherein copying includes converting the copied data values to a modified format.

15. (Original) The method of claim 14, wherein converting includes applying a case folding algorithm to the data values.

16. (Original) The method of claim 9, wherein said third data type is selected from the group consisting of said first data type, said second data type and a data type different from the first and second data types.

17. (Original) The method of claim 9, wherein when the first tenant creates a record for the first table, executing a process that determines whether the data value in the first data field for that record satisfies a threshold criteria, and if so, processing an action rule.

18. (Original) The method of claim 17, wherein the action rule indicates a recipient of a notification, the method further including automatically sending a notification message to the recipient.

19. (Original) The method of claim 9, further including defining an owner field for the first data table, wherein each data value stored in the owner field indicates an hierarchical user access level for the associated record.

20. (Previously Presented) A computer readable medium storing code for controlling a database system to store multiple fields for multiple tenants in a single multi-tenant data structure, the code comprising instructions to:

define a multi-tenant data structure having a plurality of data columns and one or more index columns;

define a first data field for a first tenant, said first field having a first data type;

define a second data field for a second tenant, said second field having a second data type, wherein the second data type may be different than said first data type;

store the data values of first and second fields to a single column in the data structure when records having data values in the first and second fields are created by the first and second tenants, wherein the single column includes data values that may include different data types for different tenants.

21. (Original) A computer readable medium storing code for controlling a database system to store multiple fields for multiple tenants in a single multi-tenant data structure, the code comprising instructions to:

define a multi-tenant data structure having a primary key column, an organization id column and a plurality of data columns;

define a first table for a first tenant, said first table having a first data field, and said first tenant having a first tenant id;

assign a first table id to the first table;

define a second table for a second tenant, said second table having a second data field, and said second tenant having a second tenant id; assign a second table id to the second table;

wherein when records are created for the first table by the first tenant, for each created record:

a) store the value of the first data field to a single data column in the data structure;

b) store the first tenant id in the organization id column; and

c) store the first table id to the primary key column; and

wherein when records are created for the second table by the second tenant, for each created record:

a) store the value of the second data field to said single data column in the data structure;

b) store the second tenant id in the organization id column; and

c) store the second table id to the primary key column; and

wherein the first and second tables of the first and second tenants are stored in the data structure.

22. (Previously Presented) A multi-tenant database system, comprising:-

a database for storing multi-tenant data objects; and

a database management process configured to:

define a multi-tenant data structure in the database, the data structure having a plurality of data columns and one or more index columns;

define a first data field for a first tenant, said first field having a first data type;

define a second data field for a second tenant, said second field having a second data type, wherein the second data type may be different than said first data type;

store the data values of first and second fields to a single column in the data structure when records having data values in the first and second fields are created by the first and second tenants, wherein the single column includes data values that may include different data types for different tenants.

23. (Original) A multi-tenant database system, comprising :

a database for storing multi-tenant data objects; and

a database management process configured to:

define a multi-tenant data structure in the database, wherein the data structure has a primary key column, an organization id column and a plurality of data columns;

define a first table for a first tenant, said first table having a first data field, and said first tenant having a first tenant id;

assign a first table id to the first table;

define a second table for a second tenant, said second table having a second data field, and said second tenant having a second tenant id;

assign a second table id to the second table;

wherein when records are created for the first table by the first tenant, for each created record:

a) store the value of the first data field to a single data column in the data structure;

b) store the first tenant id in the organization id column; and

c) store the first table id to the primary key column; and

wherein when records are created for the second table by the second tenant, for each created record:

a) store the value of the second data field to said single data column in the data structure;

b) store the second tenant id in the organization id column; and

c) store the second table id to the primary key column; and

wherein the first and second tables of the first and second tenants are stored in the data structure.

24. (Previously Presented) The method of claim 1, wherein the multi-tenant data structure comprises a relational database data structure.

25. (Previously Presented) The method of claim 4, wherein the converting includes applying a case folding algorithm to the data values.

26. (Previously Presented) The method of claim 4, wherein the modified format comprises a common data type corresponding to the index column.